Improving SVM-Linear Predictions Using CART for Example Selection

[1] Faculty of Engineering, University of Porto, Portugal
jmoreira@fe.up.pt
[2] Faculty of Economics, LIACC, University of Porto, Portugal
amjorge@liacc.up.pt
[3] Faculty of Economics, LIACC, University of Porto, Portugal
csoares@liacc.up.pt
[4] Faculty of Engineering, University of Porto, Portugal
jfsousa@fe.up.pt

Abstract. This paper describes the study on example selection in regression problems using μ-SVM (Support Vector Machine) linear as prediction algorithm. The motivation case is a study done on real data for a problem of bus trip time prediction. In this study we use three different training sets: all the examples, examples from past days similar to the day where prediction is needed, and examples selected by a CART regression tree. Then, we verify if the CART based example selection approach is appropriate on different regression data sets. The experimental results obtained are promising.

1 Introduction

The performance of prediction algorithms can be improved by selecting the examples used for training. The improvement of performance may correspond to faster training by reducing the number of examples and/or better predictions by selecting the most informative examples [1].

In this paper we approach the problem of example selection (or instance selection) mainly to improve predictive performance.

In section 2 we present results using three different training sets for μ-SVM with the linear kernel and with different parameter sets. All these experiments were done using bus trip time data. In section 3 we present the results of using CART's leaf node (one of the two example selection techniques described in section 2) on different regression data sets collected from L. Torgo's repository ([11]) and compare them with the ones from μ-SVM linear without example selection for different parameter sets. We also compare both algorithms using the corresponding best parameter set with CART and linear regression. In section 4 we discuss the results obtained and in the next section we review related work. We end with a review of the work done and a discussion of the guidelines for future work.

F. Esposito et al. (Eds.): ISMIS 2006, LNAI 4203, pp. 632–641, 2006.

2 Example Selection for Bus Trip Time Prediction

In this section we present results of example selection applied to a real decision support problem of bus trip time prediction. This problem consists in predicting three days ahead the duration of a particular urban bus trip. The predicted trip times are then used to define crew duties and reduce costs in terms of extra time payed.

We present results using μ-Support Vector Machine ([8], [7]) with the linear kernel. Tests were done using data from January 1st 2004 to March 31st 2004, i. e., 2646 trip records. The training is done on a sliding window one month long and the test data is one day long. The lag between the train and test sets is three days long.

The explanatory variables used are: (1) start trip time (in seconds); (2) day type (bank holiday, normal, bridge or tolerance); (3) weekday; and (4) day of the year. The bridge day type corresponds to a day, usually a Monday or a Friday, respectively if a bank holiday falls on a Tuesday or a Thursday, that people take as holiday so people can enjoy a four-day weekend. The tolerance day type is similar to the bridge day type, but, in the tolerance case, this day was declared by the government a day that civil servants can take off. The start trip time and the day of the year are numeric while the other two are symbolic. The target variable is the trip time duration and is a numeric one. The series is irregularly time spaced with 29,5 trips per day in average. See [6] for a more complete description of the bus trip time data set.

On these trip time data, we have tried 3 different methods for example selection:

1. All: use all the examples from the same bus line;
2. Equivalent days (ed): use the examples from the same bus line and from identical past days;
3. Leaf node (ln): use the examples from the same bus line and from the same leaf node of a CART (CART - Classification And Regression Trees [2]).

Equivalent days were defined by visual inspection and using experts knowledge. For each bus line were defined nine groups of equivalent days:

- Day Type = Normal and working days (from Monday to Friday)
- Day Type = Normal and Saturdays
- Sundays
- Day Type = bank holiday and weekday on Monday, Friday
- Day Type = bank holiday and weekday on Tuesday, Thursday
- Day Type = bank holiday and Wednesday
- Day Type = bank holiday and Saturdays
- Day Type = bridge day
- Day Type = tolerance day

The example selection using the CART leaf nodes was done by first obtaining a CART regression tree over the initial training set. To predict the time for a

new trip, we identify the leaf node of the tree where the new trip would belong and obtain all the elements from this leaf node. The members of the training set are the members from the selected leaf node. We have used the rpart (from R-project [10]) with the default parameters. We use the mean squared error (mse) as evaluation criterion.

Results for different parameter sets are presented in fig. 1. The parameter sets were obtained by combining the two dimensional parameter space (C and μ) where C ranges from $2^{(2*(\mathbf{-2}))}$ to $2^{(2*\mathbf{6})}$; and μ ranges from $\mathbf{1}/10$ to $\mathbf{10}/10$. The numbers in bold face will be referred as C index and μ index, respectively.

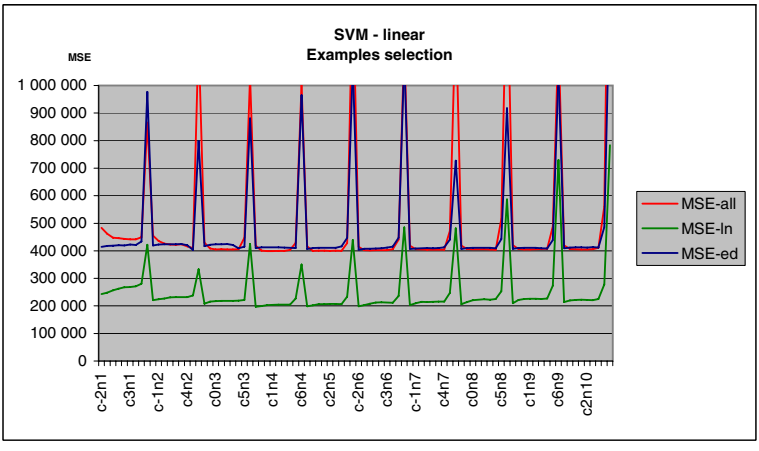


Fig. 1. Example selection for μ-SVM linear. The x-axis's values are expressed as $c[C$ index$]n[\mu$ index$]$.

From the analysis of the results we can see that the best results for all parameter sets are obtained using the leaf node method. These results were the motivation for the study that we describe in the next section.

3 Using CART's Leaf Node Members as Training Set for μ-SVM Linear on Regression Data Sets

In this section we validate if the positive results, obtained in the previous section for μ-SVM linear using the CART's leaf node approach for example selection, can be generalized to the regression problem. To run these tests we have collected eleven regression data sets from the L. Torgo's repository ([11]). We have selected data sets of diverse sizes, with the largest one, abalone, having more than 4000 examples.

The experimental procedure (figure 2, adapted from [9]) uses 10-fold cross validation. The selection task was performed by the CART's leaf node approach

described in the previous section. For each fold, the CART runs once. The parameters tuning was performed by varying C from $2^{(2*(\mathbf{-4}))}$ to $2^{(2*\mathbf{4})}$; and μ from $(2 * \mathbf{0} + 1)/10$ to $(2 * \mathbf{4} + 1)/10$. The 45 parameter sets for both training sets (with and without selection) run over the same 10 folds previously generated, i. e., we are generating paired samples. Pre-all and Pre-ln are, each one, a set of mean squared error values, one for each parameter set tested.

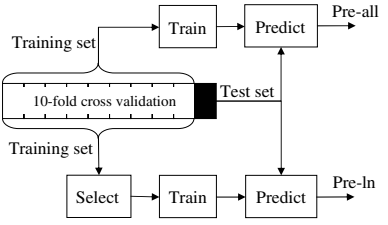


Fig. 2. Experimental procedure

This process was executed 5 times for each data set. In each of the 5 runs, different folds are obtained in 10-fold used for cross-validation. Figures from 3 to 8 present the results.

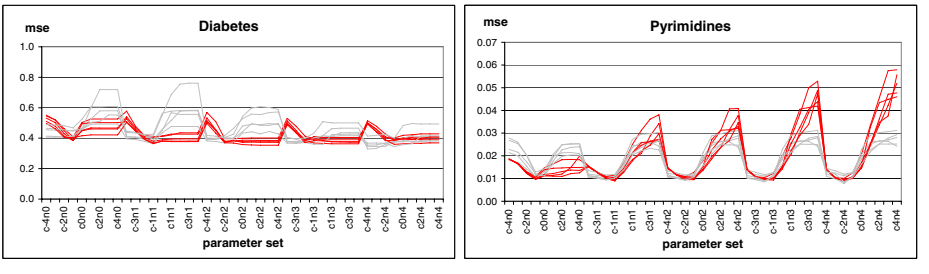


Fig. 3. Diabetes and pyrimidines data sets. The 5 dark lines represent the Pre-all results and the 5 clear lines represent the Pre-ln results. The x-axis's values are expressed as $c[C$ index$]n[\mu$ index$]$.

Figure 9 shows how much relative time is spent using the example selection technique compared with the values obtained using all data. The data sets are ordered by number of examples (ascending order). The values associated to Pre-all are calculated doing the ratio of the average time to run the 5 Pre-all results over the average time to obtain the 5 Pre-ln results.

Further tests were executed using the parameter set, from the 45 we have tested, with the best average for the 5 observations of Pre-all and the equivalent for Pre-ln. Results using the best parameter set from Pre-all and Pre-ln will be referred as all and ln respectively. We have also compared these results with

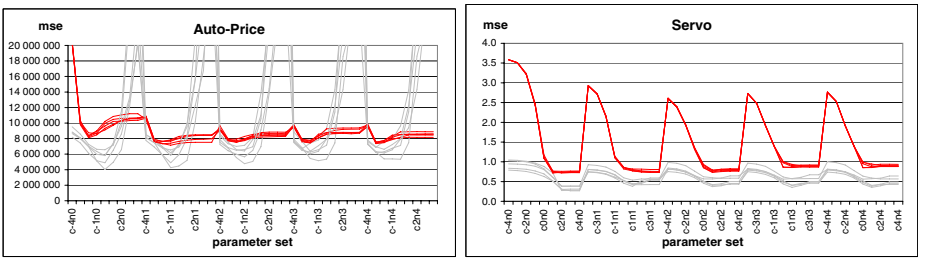


Fig. 4. Auto-price and servo data sets

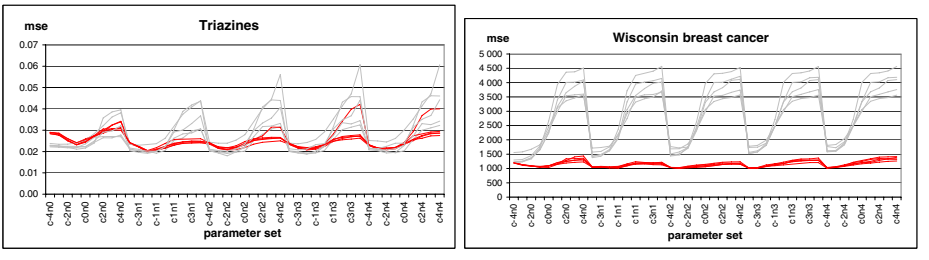


Fig. 5. Triazines and Wisconsin breast cancer datasets

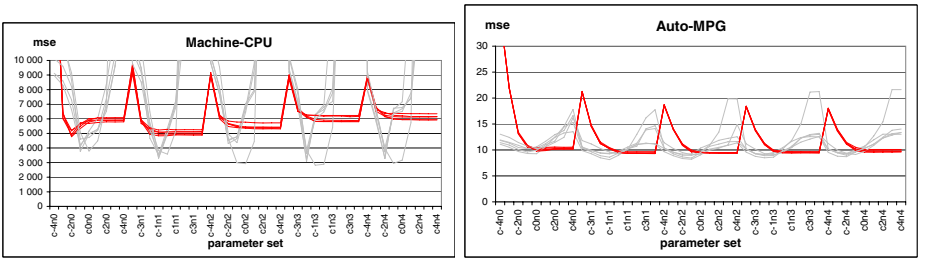


Fig. 6. Machine CPU and auto-MPG data sets

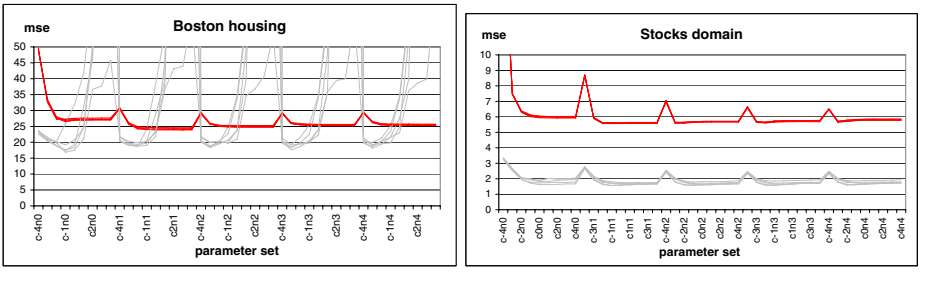


Fig. 7. Boston housing and stocks domain data sets

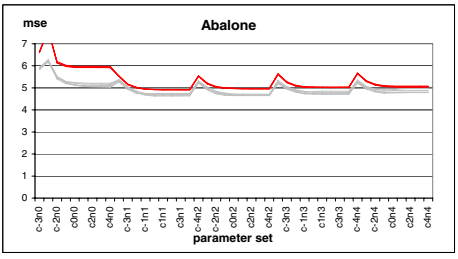
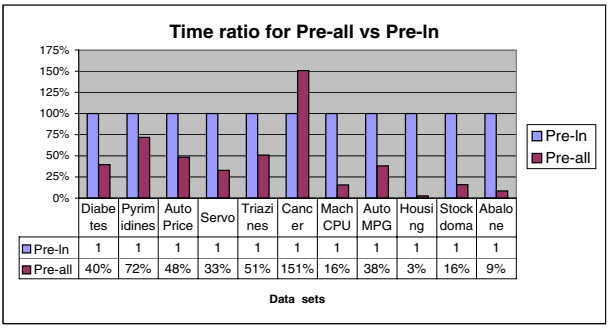


Fig. 8. Abalone data set



	Diabetes	Pyrimidines	Auto Price	Servo	Triazines	Cancer	Mach CPU	Auto MPG	Housing	Stock doma	Abalone
Pre-ln	1	1	1	1	1	1	1	1	1	1	1
Pre-all	40%	72%	48%	33%	51%	151%	16%	38%	3%	16%	9%

Fig. 9. Time ratio Pre-all vs Pre-ln

CART and linear regression. We got 50 observations for each technique using cross-validation but, this time, the samples are not paired. The lowest mse value for each data set is in bold face. We have used the default parameter set for rpart (CART) and lm (linear regression) functions from [10] (see figure 10). Lm results for the pyramidines and triazines data sets are not reported because prediction from a rank-deficient fit may be misleading. Sd represents the standard deviation. The p-value for lm, CART and all refer to the p-value obtained on a hypothesis test about the difference between that technique mean and the mean of the ln results. The alternate hypothesis is to verify if the ln mean is lower than the mean of the method being compared. We use a type I error of 5%. The p-value is in bold face when is lower than 5%, i.e, when the null hypothesis is rejected.

Figure 11 compares execution times between the four tested methods.

4 Discussing Results

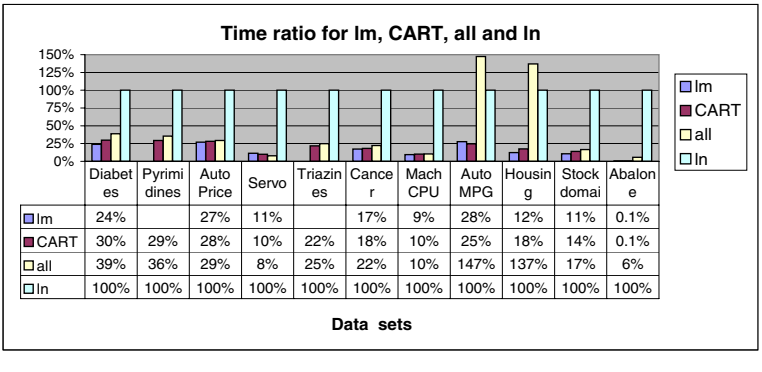
The first tests (presented in figures 3 to 9) intended to tune the parameters in order to select one parameter set for each data set. The tuning was done for two different approaches previously reported (Pre-all and Pre-ln). Some considerations must be done about these tests and their results.

- In general, results are more sensitive to C than to μ parameter values.
- The sensitivity to the parameter sets is very different between data sets and, for some data sets, is also very different between both tested techniques.
- There are several factors affecting time performance results (figure 9). In part, Pre-ln execution time is higher because of CART, even if its complexity is lower than in SVM's case. The use of CART reduces the training examples, reducing the time to train SVM. However, the number of SVM's trained has (if our algorithm was optimized, which is not the case), as upper bound, the number of the CART leaf nodes. In our case we train a new model when the value to predict falls in a different leaf node than the previous value to predict, increasing the number of trained models.
- The time to train a model depends strongly on the parameter set we are using. The higher the values of C and μ parameters, the slower is the training time. This is particularly evident in the C parameter case.

The analysis of results for Pre-all and Pre-ln was done using the Friedman rank sum test since the Kolmogorov-Smirnov Lilliefors test rejects the normality hypothesis and, consequently, it was not possible to use the ANOVA method with just 5 elements for each group. We used as blocks the eleven data sets

	lm			CART			all			ln	
Data Set	**mean**	**sd**	**p-value**	**mean**	**sd**	**p-value**	**mean**	**sd**	**p-value**	**mean**	**sd**
Diabetes	0.397	0.020	**0.2%**	0.393	0.035	**2.1%**	0.384	0.019	15.7%	**0.377**	0.044
Pyrimidines				0.0122	0.00094	**0.0%**	**0.0093**	0.00041	100.0%	0.0101	0.00158
Auto Price	7675516	363507	**0.0%**	8264533	735163	**0.0%**	7445161	169687	**0.0%**	**5215136**	787555
Servo	0.71	0.023	**0.0%**	0.77	0.073	**0.0%**	0.73	0.020	**0.0%**	**0.39**	0.120
Triazines				0.0217	0.00193	**0.0%**	0.0200	0.00032	21.7%	**0.0198**	0.00194
Cancer	1147	41.0	100.0%	1480	94.2	**0.0%**	**1020**	12.8	100.0%	1278	71.6
Mach CPU	4951	619	**0.0%**	10100	1110	**0.0%**	5001	279	**0.0%**	**4076**	1340
Auto MPG	9.2	0.159	6.2%	11.5	0.619	**0.0%**	9.4	0.159	**0.0%**	**9.1**	0.558
Housing	23.8	0.297	**0.0%**	23.3	1.240	**0.0%**	24.5	0.389	**0.0%**	**18.8**	2.183
Stock domain	5.51	0.023	**0.0%**	3.93	0.075	**0.0%**	5.60	0.029	**0.0%**	**1.64**	0.126
Abalone	4.91	0.014	**0.0%**	5.86	0.054	**0.0%**	4.91	0.011	**0.0%**	**4.69**	0.056

Fig. 10. Comparing different algorithms



	Diabetes	Pyrimidines	Auto Price	Servo	Triazines	Cancer	Mach CPU	Auto MPG	Housing	Stock domain	Abalone
lm	24%		27%	11%		17%	9%	28%	12%	11%	0.1%
CART	30%	29%	28%	10%	22%	18%	10%	25%	18%	14%	0.1%
all	39%	36%	29%	8%	25%	22%	10%	147%	137%	17%	6%
ln	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%

Fig. 11. Time ratio for lm, CART, all and ln

Table 1. Friedman rank sum test partial results

Alg	ps	Ri	Alg	ps	Ri	Alg	ps	Ri	Alg	ps	Ri	Alg	ps	Ri	Alg	ps	Ri
ln	c-1n1	173	all	c-1n2	357	all	c-1n3	420	all	c-3n2	518	all	c0n0	587	ln	c3n1	682
ln	c-2n3	176	all	c-2n2	368	ln	c-4n3	423	all	c0n4	519	all	c2n4	588	all	c-4n4	683
ln	c-2n4	195	ln	c-1n0	370	all	c-2n3	425	all	c-3n3	520	all	c-2n0	597	all	c-4n3	686
ln	c-2n2	197	all	c1n1	380	all	c3n2	431	all	c-1n0	523	all	c3n4	603	ln	c3n4	696
ln	c-1n2	210	all	c0n2	381	all	c4n2	443	ln	c1n0	523	ln	c-4n0	604	all	c-4n2	700
ln	c-2n1	229	ln	c-3n1	381	ln	c0n4	445	all	c2n3	531	all	c2n0	605	ln	c3n3	702
ln	c-1n3	247	all	c1n2	386	all	c-1n4	460	all	c3n3	536	all	c4n4	613	ln	c3n0	706
ln	c-1n4	268	ln	c0n0	392	ln	c-4n2	462	all	c4n3	549	ln	c2n1	625	ln	c4n1	706
ln	c-3n3	293	all	c2n1	395	all	c-2n4	465	all	c-3n1	550	ln	c2n2	625	ln	c4n2	709
all	c-1n1	297	ln	c0n2	401	all	c0n3	475	ln	c1n2	555	all	c3n0	627	ln	c4n4	721
ln	c-3n4	312	all	c3n1	402	ln	c-4n1	492	all	c1n4	566	all	c4n0	637	ln	c4n3	727
ln	c0n1	322	all	c4n1	407	ln	c-2n0	493	ln	c1n3	568	ln	c2n0	646	all	c-4n1	729
ln	c-3n2	330	ln	c0n3	413	all	c1n3	500	ln	c-3n0	569	ln	c2n3	646	ln	c4n0	740
all	c0n1	345	ln	c-4n4	416	ln	c1n1	508	all	c1n0	582	ln	c2n4	650	all	c-3n0	746
all	c-2n1	357	all	c2n2	417	all	c-3n4	516	ln	c1n4	584	ln	c3n2	674	all	c-4n0	817

tested, and as groups the combination of the algorithm and the parameter set used (2 algorithms $\times$ 45 parameter sets). As observation we used the average of the 5 existing observations in each group. The ranking table is presented in table 1 where Ri is the rank sum for the 11 data sets. The null hypothesis for this test is that the group effect is equal for all the groups. The null hypothesis is rejected with p value = 0.00%. This result allows us to say that there are meaningful statistical differences between the different algorithms / parameter sets.

We also observe, in Table 1, that the 9 top ranking models were obtained using the ln approach. It is also a tendency (although with exceptions) that for the same set of parameters, the model obtained with the ln example selection strategy ranks higher than the model obtained with the all strategy.

We have then compared results of four different algorithms: the two previously discussed with the best parameter set pre-selected for each data set, the CART and the linear regression. Some considerations must be done about these tests and their results.

- Time execution for ln and all (figure 11) are not equivalent to previous results (see figure 9) for all data sets as, for example, the Wisconsin breast cancer, auto-MPG and Boston housing data sets. This happens when the best parameter set for each one of the methods are very different, in particular for the C parameter. As we have previously said, the execution time increases strongly for higher parameter values. This is particularly true for the C parameter.
- With the exception of the pyrimidines and the Wisconsin breast cancer data sets, the best results using CART's leaf node members as training set are similar or better than without example selection.

- For the data sets auto-price, servo, machine-CPU, Boston housing, stocks domain and abalone results using example selection (ln) are meaningfully better compared to all other techniques. For the diabetes and triazines data sets the results are meaningfully better for ln comparing to all other techniques except for μ-SVM linear without example selection (p values of 15.7% and 21.7%, respectively). For the auto-MPG data set the ln method is not statistically better than the linear model (p value of 6.2%) but it is for the other two methods (all and CART). Ln performs badly for the Wisconsin breast cancer data set (looses for the all and the linear models) and also for the pyrimidines data set (looses for the all method).
- Ln performs always better than CART. This can be explained by the fact that, in spite of the similarity of the approaches, the prediction model is the average in CART's case and the μ-SVM linear in ln's case (as expected, μ-SVM linear performs better than the average for all data sets).
- Ln variance is much higher than all variance. This can be explained by CART induction algorithm (compares standard deviation values between ln and CART) and also by the reduction of training data in ln method.

5 Related Work

In this paper we present a study on example selection in order to improve predictive accuracy ([1]). Other authors use example selection in order to reduce training time ([5]). The works on example selection and feature selection have many aspects in common ([1]). Techniques for feature selection are usually classified as filters or wrappers ([4]). Filter techniques select the features independently of the used induction algorithm while the wrapper approaches have an embedded evaluation of the selection according to the used induction algorithm ([4]). Using this classification we can say that our approach is a filter one. A similar approach had been tried before in [9]. In that work, the motivation was to test if the use of the support vectors from SVM as training set is an example selection technique that is model independent, i. e., if it can improve results independently of the prediction algorithm chosen. The results obtained in [9], however, are not as expected, since none of the used algorithms (multi-layer perceptrons, nearest neighbors and the C4.5 algorithm for decision trees) improves predictions. Moreover, different algorithms obtain different results. We believe that filter techniques are not model independent, i.e., there is no guarantee that the best training set for a certain induction algorithm is the best one for another induction algorithm. However, we believe that a wrapper approach may be model independent. The wrapper approach has the additional advantage of giving better results than the filter approach. It has the disadvantage of being computationally heavier. Both filter and wrapper approaches are worth researching in the future.

The use of decision trees for feature selection is known ([3]), but we do not know any work using decision trees for example selection.

6 Conclusions and Future Work

This paper describes a study on example selection for μ-SVM linear. Firstly we describe two approaches for example selection on a bus trip time prediction task. We have done tests using μ-SVM with the linear kernel. We observed the impact of example selection for this data set. Then, we tried using as training examples for μ-SVM linear the ones on the same CART's leaf node as the case being predicted. This technique was then tested on other 11 regression data sets. Results show that, with the exception of two data sets, all the others give equal or lower mean squared error when using the example selection technique. This is a very promising result.

Future work will include the reduction of execution time by optimizing our algorithm and by testing faster decision trees induction algorithms, the repetition of these tests for other SVM kernels (we also got promising results using μ-SVM with the gaussian and the sigmoidal kernels for the bus trip time data set), and the tuning of the decision tree induction algorithm parameters.

Acknowledgments

This work was partially supported by FCT - Fundação para a Ciência e a Tecnologia, project reference: POCT/TRA/61001/2004 and FEDER e Programa de Financiamento Plurianual de Unidades de I&D.

References

1. A. L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2):245–271, 1997.
2. L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and regression trees*. Chapman and Hall/CRC, 1984.
3. C. Cardie. Using decision trees to improve case-based learning. In *10th International conference on machine learning*, pages 25–32. Morgan Kaufmann, 1993.
4. R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.
5. H. Liu and H. Motoda. On issues of instance selection. *Data Mining and Knowledge Discovery*, 6(2):115–130, 2002.
6. J. M. Moreira, A. Jorge, J. F. Sousa, and C. Soares. Trip time prediction in mass transit companies. a machine learning approach. In *10th EWGT*, pages 276–283, 2005.
7. B. Scholkopf, A. J. Smola, R. Williamson, and P. Bartlett. New support vector algorithms. Technical Report NC2-TR-1998-031, 1998.
8. A. J. Smola and B. Scholkopf. A tutorial on support vector regression. Technical Report NC2-TR-1998-030, 1998.
9. N. A. Syed, H. Liu, and K. K. Sung. A study of support vectors on model independent example selection. In *5th ACM SIGKDD*, pages 272–276, 1999.
10. R. D. C. Team. R: A language and environment for statistical computing. Technical report, R Foundation for Statistical Computing, 2004.
11. L. Torgo. Regression data repository, http://www.liacc.up.pt/ ltorgo.